

Redefining decay files

N, G, I

MIT meeting, March 13, 2003

- what for
- procedure
 - ✓ proof
 - ✓ algorithm
- script

- ♦ Purpose:
 - ✓ produce clean MC samples
 - ✓ reduced generation time and sample sizes
- ♦ How:
 - ✓ overwrite relevant particle decays
 - ✓ include only decay chains we are interested in for reconstruction
- ♦ Why is it incorrect to just remove decay lines
 - ✓ relative probability for decay chains becomes inconsistent!
 - ✓ need to re-calculate particle branching ratios...

Procedure

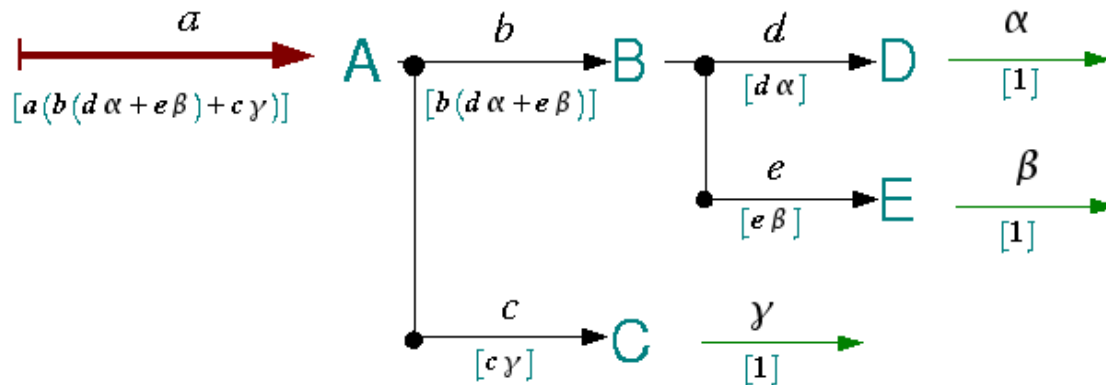
- Select decays
 - ✓ from full decay file select relevant decays
- Rename decay particles involved
 - ✓ this ensures only the particles involved in the full decay chain are redefined
- Multiply BRs recursively, backwards along the decay chain
 - ✓ this ensures chain relative probabilities are preserved
- Normalize BR values
 - ✓ for each particle, divide the 'updated' BR values of its selected daughters by the respective sum
 - ✓ also done automatically by decay package

Proof of algorithm

Compare ratio of probabilities of chains ABD → and AC →

Before:
$$\frac{P(ABD)}{P(AC)} = \frac{a b d \alpha}{a c \gamma}$$

After:
$$\frac{P(ABD)}{P(AC)} = \frac{a' b' d' \alpha'}{a' c' \gamma'} = \frac{b' d' \alpha'}{c' \gamma'} = \frac{\frac{b(d\alpha + e\beta)}{b(d\alpha + e\beta) + c\gamma} * \frac{d\alpha}{d\alpha + e\beta} * \frac{\alpha}{\alpha}}{\frac{c\gamma}{b(d\alpha + e\beta) + c\gamma} * \frac{\gamma}{\gamma}} = \frac{bd\alpha}{c\gamma}$$



Perl implementation

Module examples:

Displaying decay chains →

Redefining branching ratios ↓

```
U
sub redefine_dec_br{
  for ($i=0; $i<=$#decpart_order; $i++) {
    $p=$decpart_order[$i][0];
    $p_=$decpart_order[$i][1];
    $dsbr=sum_bratio($p);
    for ($j=0; $j<=$#decpart_order; $j++) {
      $part=$decpart_order[$j][0];
      $part_defr="dec_defr$part";
      foreach $n (@$part_defr) {
        my $n1=$n;
        if ( $n1 =~ "CHANNEL" ) {
          if ( $n1 =~ $p_ ) {
            my @chn = split(/ +/, $n1);
            my $br_old=$chn[2];
            my $br_new=$br_old*$dsbr;
            $n =~ s/$br_old/$br_new/;
          }
        }
      }
    }
  }
}
```

```
sub print_chain {
  my $i=$_[0];
  my $j=$_[0];
  my $pt=$_[1];
  print OFC "\n\n *****\tPrinting $pt decay chains...\n\n" if $i==0;
  my $num=$pdaughter{$pt}[0][0]+1;
  print OFC "      $pt\t\t \n" if $i==0;

  for $i0 ( 1 .. $num) {
    my $j=$i;$j++;
    my $num_p0=$pdaughter{$pt}[0][$i0]+1;
    my $dau1="@{ $pdaughter{$pt}[$i0] }";
    if ( $pt =~ "B0" ) {$ddau1=$pdaughter{$pt}[$i0][3];}
    else {$ddau1=$pdaughter{$pt}[$i0][1];}
    my $numd1=$pdaughter{$ddau1}[0][0]+1;
    if ($i0==1) {
      print OFC "\t","  |\t\t" x $i, "---o--->\t$dau1\n";
    } elsif ($i0<$num) {
      if ($i==2) { print OFC "\t","  |\t\t" x $i, " |--->\t$dau1\n";}
      else { print OFC "\t","  |\t\t" x $i, " |--->\t$dau1\n";}
    } else {
      if ($i==2) {print OFC "\t","  |\t\t" x $i, " |--->\t$dau1\n";}
      else {print OFC "\t","  |\t\t" x $i, " |--->\t$dau1\n";}
    }
  }

  print_chain ($j,$ddau1) if ( $ddau1 !~ 'K\+' && $i<20);
}

print OFC "\n\n\n\n" if $i==0;
print OFC "\t |\t\t \n" if $i==1;
print OFC "\t |\t\t |\n" if $i==2;
}
```